

App'n. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

Remarks/Arguments

These remarks are in response to the Office Action dated June 6, 2005. This reply is timely filed.

At the time of the Office Action, claim 1 was cancelled. Claims 2-18 were pending in the application. Claims 2, 10 and 18 were objected to because of informalities. Claims 2, 5-10 and 13-18 were rejected under 35 U.S.C. §103(a). The rejections are set out in more detail below.

I. Claim Objections

Claims 2, 10 and 18 were objected to based on minor informalities. In particular the Examiner notes that the recitation of "said re-synchronized interprocess communication link" in claims 2 and 10, and "said established interprocess communications link" in claim 18 are lacking of antecedent basis. In response, the referenced claims have now been amended to correct this informality.

II. Brief Review of Applicants' Invention

Applicants' invention relates to a distributed messaging system for transmitting topical messages from data publishers to data consumers. In the distributed messaging system, interprocess communications between data publishers and data consumers can be re-synchronized in the event that data communications between such entities are lost. The architecture of the messaging system includes a message topic server, a plurality of message routers, and a plurality of message adapters distributed across several computing devices in a computer communications network. The message adapters are communicatively linked to applications, each of which can be a data consumer or a data publisher executing in a computing device. The data consumers consume and process data messages published by data publishers.

The data publishers can be communicatively linked to corresponding message adapters executing in the same computing device. Similarly, data consumers can be linked to message adapters executing in the same computing device. Further, each message adapter can be linked to a message router, also executing in the same computing device. Finally, each message router can be communicatively linked to a

{00006679;}

Appln. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

message topic server, although the message topic server typically executes in another computing device.

The message topic server can contain a list of message topics to which data consumers can subscribe. In addition, the message topic server also can keep a list of data publishers which publish data messages consonant with the message topics. After registering a message topic, a data publisher can provide correlating data messages to those data consumers which choose to subscribe to the message topic.

Through its associated message router, a data consumer can subscribe to a message topic which has been published by a data publisher. The message topic server can respond to such a subscription request by transmitting to the associated message router a host identification of a second message router from which the requesting data consumer can receive the data messages to which it has subscribed. Notably, an interprocess communications link then can be established between the message router that is associated with the data consumer and the second message router. Messages consonant with the requested message topic then can be transmitted over the interprocess communications link.

Notably, the distributed messaging system of the present invention can utilize a shared state memory which stores both message traffic and network configuration data. More particularly, the three network components (message adaptor, message router and message topic server) can form the shared state memory. In consequence, the message traffic and network configuration data is readily available for reconstruction and re-synchronization of interprocess communications should the distributed messaging system experience a communications fault with any combination of the three network components. Moreover, by virtue of the shared state memory architecture of the present invention, recovery and re-synchronization processes can be implemented without loss of data.

III. Claim Rejections on Art

Claims 2, 5-10 and 13-18 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,298,455 to Knapman, et al. ("Knapman") in view of

{00006679;}

Appl. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

U.S. Patent No. 6,070,191 to Narendran, et al. ("Narendran"), and in further view of U.S. Patent No. 6,662,206 to Banavar et al. ("Banavar").

a. Brief Review of Cited Art

Knapman discloses a technique for carrying out "failover." In failover, should a first distribution agent fail, other distribution agents that communicate directly with the first distribution agent will transfer their subscriptions to a second distribution agent which is a sibling of the first distribution agent. Knapman implements failover by providing a data processing broker network having a plurality of broker data processing apparatuses. Each of the apparatuses is configured to assign a broker-specific sequence number to a received message. A first of the broker apparatuses has a software unit for determining a failure of a neighboring broker apparatus which has provided published messages on a first topic. The first broker apparatus also includes a software unit for sending historic resubscriptions with respect to the first topic to each antecedent broker apparatus of the failed neighboring broker apparatus. The historic resubscriptions are sent using the broker-specific sequence number corresponding to each antecedent broker apparatus.

Nerendran discloses a server system for processing client requests received over the Internet. The server system includes a cluster of N document servers and at least one redirection server. The redirection server receives a client request from the network and redirects it to one of the document servers, based on a set of pre-computed redirection probabilities. Each of the document servers may be an HTTP server that manages a set of documents locally and can service client requests only for the locally-available documents. A load distribution algorithm is used to distribute a set of documents across the document servers. In the event of a server failure, the redirection probabilities are recomputed and a URL of a new server is provided to clients requesting documents contained on the failed server.

Banavar discloses a method for delivering to a subscriber node an optimized version of an event message stream. The optimized version of the event message stream is essentially a summary of the event message stream based on an event interpretation rule defined by the subscriber node. The optimized version of the event

{00006679;}

Appln. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

message stream is useful for recovering from a network interruption. Banavar monitors the event stream and then applies the interpretation rules for a subscriber node to generate the smaller optimized version of the event stream.

In Banavar, a sequencer keeps a uniform total order of messages common to all subscribers, even during periods when subscribers are disconnected. The state of each subscriber is updated as a function of the event message stream. The state log for each subscription represents all the successive states through which the respective subscriber has passed. If a subscriber disconnects from the network and later reconnects, the sequencer uses the state log to determine an optimized version of the event message stream to achieve a desired goal state for the subscriber. This optimized version of the event message stream is communicated to the data subscriber using the network communication link.

b. Claim Rejections

Claims 2, 10 and 18 each recite sharing state memory among at least a message topic server, a first message router and a second message router to store both message traffic and network configuration data. Responsive to a communication fault in the first message router, the second message router, the data consumer or the message server, the interprocess communications connection is re-synchronized from the shared state memory, which provides network configuration data and network traffic data. Importantly, the re-synchronization process can be implemented without loss of data.

In rejecting claims 2, 10 and 18, the Examiner concedes that Knapman does not disclose sharing of state memory or resynchronizing interprocess communications as claimed. However, the Examiner contends that these deficiencies are remedied by the combination of Narendran and Banavar. Applicants respectfully disagree.

Narendran discloses a simple load balancing mechanism in a client server context. A redirection server redirects client requests using a load distribution algorithm. The load distribution algorithm is used to equalize the rate at which documents are accessed on a plurality of document servers. In the event of a

{00006679;}

Appln. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

document server failure, the redirection server re-computes the loading and rebalances requests among the remaining document servers.

Narendran's load balancing scheme has little or no relevance with regard to Applicants' claimed invention. Narendran does not disclose or suggest the shared state memory or resynchronization of messages as recited in Applicants' claims. For example, Narendran's redirection mechanism resides only in the redirection server. The redirection information is not shared among multiple message routers, data consumers and message servers as recited in Applicants' claims. Moreover, it is apparent that Narendran's redirection process is not equivalent to Applicants' claimed resynchronization of interprocess communications. Redirecting documents requests to different servers is not equivalent to re-synchronizing a network. Finally, Narendran does not even concern interprocess communications. Instead, Narendran describes a conventional client/server arrangement using standard TCP/IP communication protocols. See Col. 3, lines 39-45.

Banavar fails to make up for the deficiencies of Narendran and Knapman. Banavar uses a sequencer to monitor a stream of messages to a subscriber. The sequencer applies an interpretation rule to generate a smaller optimized version of the message stream that can be communicated to a particular node when communications are resumed.

The Examiner asserts that Banavar discloses the use of state memory to store both message traffic data and network configuration data, and re-establishing a connection from the state memory after recovering from a communication fault. Specifically, the Examiner cites (Fig. 2 and col. 7, lines 22-63). However, Banavar's state memory contains information about a subscriber's state, not the network configuration. See col. 7, lines 50-64. The so called "state log" is used to determine an optimized version of the event message stream to achieve a desired goal state for the subscriber.

Aside from these obvious distinctions, Banavar also fails to show that the so called state log described therein is shared among (1) a message topic server, (2) a first message router associated with a data consumer, and (3) a second message router associated with a data publisher. In fact, none of the cited references disclose this

{00006679;}

Appln. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

feature. Sharing of state memory among the distributed elements, as claimed by Applicants, is important in order to ensure resynchronization regardless of which network nodes may suffer a failure. In the claimed invention, resynchronization can be performed after a failure or interruption at any network node. Thus, even if Banavar's sequencer did contain network configuration data, a single fault in Banavar's sequencer would leave Banavar's system highly vulnerable and unable to re-synchronize communications. That is because the state log information for a subscriber is not shared as claimed by Applicant.

Notwithstanding the foregoing, the Examiner contends that it would have been obvious to one having ordinary skill in the art to have utilized Banavar's state memory to reestablish communication connections in Banavar. Applicants disagree. The cited combination of references lack important features of the claimed invention that are neither taught nor suggested. Banavar's state memory merely contains information about a subscriber's state. See col. 7, lines 50-64. It does not contain overall network configuration data as claimed by Applicants and the information is not shared among nodes as claimed. Instead, Banavar's merely contains a state table for a particular subscriber that allows the sequencer to generate a summary version of a message stream. In view of the foregoing, Applicants respectfully submit that claims 2, 10 and 18 are not obvious based on the combination of Knapman, Narendran, and Banavar. The remaining dependent claims are believed to be allowable at least by virtue of their dependence upon these allowable base claims.

IV. Conclusion

Applicants have made every effort to present claims which distinguish over the prior art, and it is believed that all claims are in condition for allowance. Nevertheless, Applicants invite the Examiner to call the undersigned if it is believed that a telephonic interview would expedite the prosecution of the application to an allowance. In view of the foregoing remarks, Applicants respectfully requests reconsideration and prompt allowance of the pending claims.

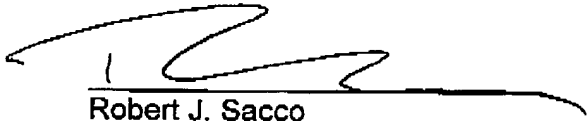
{00006679;}

Appln. No. 09/677,072
Amdt. dated: September 2, 2005
Reply to Office Action dated June 6, 2005

No fee is believed to be due under this Amendment. However, please charge any deficiencies of credit any overpayment to Deposit Account No. 50-2884.

Respectfully submitted,

9-2-05
Date


Robert J. Sacco
Registration No. 35,667
SACCO & ASSOCIATES, P.A.
P.O. Box 30999
Palm Beach Gardens, FL 33420-0999
Tel: 561-626-2222

{00006679;}